Voxelbuild: A Minecraft-Inspired Domain for Experiments in Evolutionary Creativity

L. B. Soros, Justin K. Pugh, and Kenneth O. Stanley
Department of Computer Science, University of Central Florida
Orlando, Florida 32816-2362
lisa.soros@gmail.com,jpugh@cs.ucf.edu,kstanley@cs.ucf.edu

ABSTRACT

The fields of artificial life and evolutionary robotics have seen growing interest in evolution as a source of creativity, as opposed to a tool for optimization. New intentionally divergent algorithms such as novelty search with local competition (NSLC) and MAP-Elites accordingly attempt to harness evolution's aptitude for divergence in a new search paradigm called *quality diversity (QD)*, which aims to find a wide variety of possible solutions spread across a *behavior space*. To date, QD has mainly been studied in domains where potential diversity is limited. In anticipation of future, more open-ended applications of QD algorithms, this paper introduces a new domain inspired by the popular Minecraft video game featuring a larger behavior space that is substantially more difficult to exhaust. Preliminary results are presented, showcasing sample block structures built by evolved neural network controllers.

CCS CONCEPTS

•**Computing methodologies → Artificial life; Evolutionary robotics;** *Generative and developmental approaches;*

KEYWORDS

Quality diversity, open-ended evolution, neuroevolution

ACM Reference format:
L. B. Soros, Justin K. Pugh, and Kenneth O. Stanley. 2017. Voxelbuild: A Minecraft-Inspired Domain for Experiments in Evolutionary Creativity. In *Proceedings of GECCO '17 Companion, Berlin, Germany, July 15-19, 2017,* 2 pages.
DOI: http://dx.doi.org/10.1145/3067695.3075605

1 INTRODUCTION

Evolutionary algorithms have long been applied to optimization problems. However, a new class of intentionally divergent algorithms called quality diversity (QD) algorithms [4] (also called illumination algorithms [3]) innovate broadly, as evolution in nature has done, instead of converging to predefined target(s). Maze navigation tasks have emerged as a popular benchmark domain for experiments with these new algorithms[1, 4]. Although simple, they serve as computationally inexpensive proxies for complex physical control tasks. However, because the space of possible innovations in such domains is inherently limited (to the possible paths through the maze), these domains offer limited insight into evolution's capacity for open-ended innovation. Locomotion tasks [6] are arguably more open-ended than maze navigation tasks because agents are usually tasked with the more general goal of walking as far as possible rather than reaching a target point, which increases the range of viable behaviors. Even so, in robot locomotion, just as in maze navigation, the entire behavior space can easily be fully explored in a single run, rendering these domains inadequate for studying the full creative potential of QD algorithms.

This paper introduces a new domain for studying QD that features a behavior space that is several orders of magnitude larger and that includes regions that are more challenging to explore, compared to current iterations of maze navigation and robot locomotion. This new domain, called *Voxelbuild*, is inspired by the popular Minecraft[1] video game. Minecraft is characterized by a large open world with no explicit objectives; the player simply exists in a procedurally generated landscape of blocks (each about the size of the player) that can be rearranged to create complex structures of arbitrary scale. Human players have constructed an impressive variety of artifacts, from rudimentary shelters to enormous block-based digital circuit systems. The hope for Voxelbuild is that divergent or creativity-oriented search algorithms (e.g. QD algorithms) can eventually evolve AI controllers that build comparable artifacts. Compared to previous QD domains where the goal is to find the best possible solution at all parts of the behavior space, the focus in Voxelbuild is instead to open-endedly discover strategies for building new and interesting artifacts.

2 VOXELBUILD

As an experimental platform, Voxelbuild is designed to be computationally efficient while maintaining a high potential for subjectively interesting complexity (such as that seen in human-controlled Minecraft). In Voxelbuild, the agent is controlled by an artificial neural network (ANN) evolved by the HyperNEAT algorithm [5], instead of by a human player. To support efficient and deterministic simulation, Voxelbuild intentionally differs from Minecraft in a few ways: (1) time is divided into discrete ticks where agents perform one action per tick, (2) movement actions are simplified so that agents can only turn at 90 degree angles and move one unit forward or backward at a time, (3) agents can sense all blocks within a 5-block radius (i.e. they have a 360 degree field of view) as well as place and remove blocks in the same 5-block radius, (4) agents can only place and remove a single type of block, and (5) block placement and removal is not restricted by line-of-sight (however, blocks can only be placed adjacent to other blocks).

GECCO '17 Companion, Berlin, Germany
 978-1-4503-4939-0/17/07…$15.00
DOI: http://dx.doi.org/10.1145/3067695.3075605

[1] ©Mojang, 2011

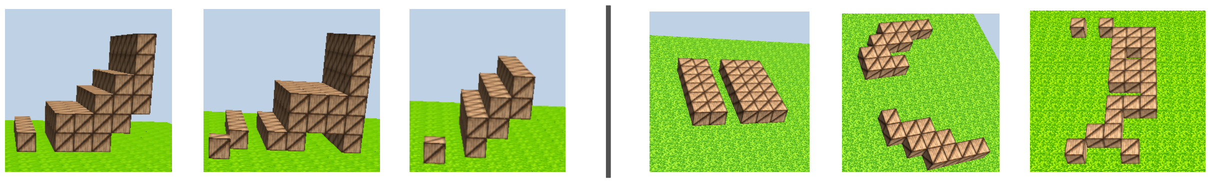

Figure 1: Structures in Voxelbuild built by evolved agents during two separate runs of novelty search with local competition (NSLC). While the structures found in these preliminary experiments are not yet as complex as those built by human Minecraft players, they suggest the potential for impressive constructions in the future.

Agent neural networks take as input an $11 \times 11 \times 11$ sensor grid that senses blocks within a 5-block radius. Each node in the 1,331-neuron input layer then contains an integer value corresponding to the block type at each location within the sensor radius (1 – crate or grassy block, 0 – boundary block, -1 – no block). Additionally, agent networks contain six effector neurons that decide which action to take on each tick (move forward, move backward, turn left, turn right, place block, or remove block) in addition to an $11 \times 11 \times 11$ output neuron matrix corresponding to locations within the 5-block radius where the selected action will be directed. If an agent requests an illegal action (e.g. placing a block in a location where there is already a block, or attempting to move through a solid block), the trial is immediately terminated, forcing agents to learn to operate within the physical constraints of the world. Two distinct $3 \times 3 \times 3$ hidden layers, placed side by side to separately process actions and action locations, connect the input and output layers.

Voxelbuild is a framework for intelligent agents to build a nearly limitless variety of block structures, providing an open-ended testbed for QD algorithms or other approaches to computational creativity. The space of possible structures is limited only by the size of the world and the duration of the trial. Importantly, larger and more complicated structures are more challenging to create because the agent must construct them without violating the rules of the world. For example, structures more than six blocks tall require that the agent build and make use of block-based scaffolding.

3 RESULTS & DISCUSSION

In this paper, Voxelbuild agents are evolved with the novelty search with local competition (NSLC) algorithm [2], designed to search for a diversity of solutions while simultaneously optimizing within local niches. NSLC, like other QD algorithms, requires both a fitness function (for computing quality) and a behavior characterization (for computing diversity). In these experiments, fitness is the net number of blocks placed by the agent multiplied by the height of the tallest block. This way, building larger and taller structures (which proved difficult in preliminary experiments) is easily and indirectly incentivized without defining explicit objectives. Finally, with the goal of remaining as objective-agnostic as possible, the behavior characterization is simply a binary vector whose values specify whether or not a solid block exists at each location in the world.

Examples of structures discovered in two different runs are shown in figure 1. While each run discovers its own diversity, the results between runs are themselves qualitatively different, suggesting the breadth of possibilities. As of this writing, no run has been able to produce an agent capable of building structures more than six blocks tall, highlighting the difficulty of accessing more advanced regions of the behavior space.

Voxelbuild offers a fun QD-oriented experimental platform that complements traditional maze navigation and robot locomotion domains. As demonstrated by these preliminary experiments, the new domain offers a unique tool with which to explore evolution's potential for open-ended creativity. While the structures found thus far are not yet as complex as those built by human Minecraft players, they display features such as symmetry and repetition that form the basis of complex structures. Current and future experiments will investigate what mechanisms enable the construction of arbitrarily complex artifacts.

ACKNOWLEDGMENTS

Special thanks to undergraduates Rafaela Frota, Michael Myerson, and Kevin Negy for help conceptualizing the domain. Additionally, this work was supported by the National Science Foundation under grant no. IIS-1421925. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

[1] Joshua E. Auerbach, Giovanni Iacca, and Dario Floreano. 2016. Gaining Insight into Quality Diversity. In *Proceedings of the 2016 on Genetic and Evolutionary Computation Conference Companion (GECCO '16 Companion)*. ACM, New York, NY, USA, 1061–1064.

[2] Joel Lehman and Kenneth O. Stanley. 2011. Evolving a Diversity of Virtual Creatures Through Novelty Search and Local Competition. In *Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation (GECCO '11)*. ACM, New York, NY, USA, 211–218.

[3] Jean-Baptiste Mouret and Jeff Clune. 2015. Illuminating search spaces by mapping elites. *CoRR* abs/1504.04909 (2015).

[4] Justin K. Pugh, Lisa B. Soros, and Kenneth O. Stanley. 2016. Quality Diversity: A New Frontier for Evolutionary Computation. *Frontiers in Robotics and AI* 3 (2016), 40.

[5] Kenneth O. Stanley, David B. D'Ambrosio, and Jason Gauci. 2009. A Hypercube-Based Indirect Encoding for Evolving Large-Scale Neural Networks. *Artificial Life* 15, 2 (2009), 185–212.

[6] Danesh Tarapore, Jeff Clune, Antoine Cully, and Jean-Baptiste Mouret. 2016. How Do Different Encodings Influence the Performance of the MAP-Elites Algorithm?. In *Proceedings of the Genetic and Evolutionary Computation Conference 2016 (GECCO '16)*. ACM, New York, NY, USA, 173–180.